AD-A085 575

May 1980

Invertible Binary Matrices

by Thomas C. Jedrey

U.S. Army Electronics Research
and Development Command
Harry Diamond Laboratories
Adelphi, MD 20783

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: HDL-TM-80-13	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Invertible Binary Matrices	5. TYPE OF REPORT & PERIOD COVERED: Technical Memorandum
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Thomas C. Jedrey	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Harry Diamond Laboratories, 2800 Powder Mill Road, Adelphi, MD 20783	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: DA: 1L161101A91A; Program Ele: 61101A
11. CONTROLLING OFFICE NAME AND ADDRESS: U.S. Army Materiel Development and Readiness Command, Alexandria, VA 22333	12. REPORT DATE: May 1980
	13. NUMBER OF PAGES: 16
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

DRCMS Code: 61110191A001 HDL Project: A10952

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Digital network synthesis
Karpovsky
Invertible binary matrices
Haar functions

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In an alternative approach to digital network synthesis, M. G. Karpovsky (1976) presents an algorithm that minimizes the number of nonzero coefficients in a Haar expansion of a Boolean function. In this algorithm, an invertible modulo two matrix must be generated subject to certain constraints. Two algorithms are presented that can generate this matrix. A simple method of generating invertible modulo two matrices is also presented.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

CONTENTS

FIGURES

1. INTRODUCTION

Current methods of designing digital devices become extremely complex for large quantities of input-output (I/O) bits. Both the Quine-McClusky and iterated consensus algorithms are impractical for input bit sizes much larger than 10 and output sizes greater than 1. An alternative approach to digital network synthesis, applicable to devices requiring large I/O bit sizes, has been proposed by M. G. Karpovsky.[1] In the methods developed, digital networks (realizing logical functions) are represented by a system consisting of a basis generator (where either Haar or Walsh functions form the basis), a memory containing expansion coefficients ("spectra"), a multiplier, and an adder.[1] For realization of systems of Boolean functions, the multiplier is not really necessary, since the basis functions assume the values 0 and ±1, or ±1. Because of the properties of Haar coefficients, Haar expansions are preferable to Walsh expansions. The local dependence of each Haar coefficient on the original function allows one to decrease the number of nonzero coefficients in the expansion of the function. This is not possible with Walsh expansions, since the Walsh coefficients depend on the values of the function over its entire domain of definition. Nonzero coefficients in a Haar expansion are reduced by a linear transformation. Karpovsky presents an algorithm that produces the optimal linear transformation of the arguments (Ch. 3, p. 115[1]); i.e., the Haar series of the function on the transformed arguments contains a minimum number of terms. In this algorithm, a matrix subject to certain constraints must be generated. It is the object of this paper to present a method for the production of this matrix.

Requirements on S.—The matrix to be generated (S) is subject to two constraints. First, it must be invertible and second, the equation

$$S \otimes \vec{T} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix} \qquad (1)$$

must be satisfied, where $\otimes$ represents ordinary matrix multiplication with all sums and products carried out modulo two,* and $\vec{T}$ is a binary vector related to the function being synthesized. Before a matrix can be produced which satisfies equation (1), it is necessary to examine a method of generating invertible modulo two matrices.

Invertible Modulo Two Matrices.—An arbitrary square matrix is nonsingular if the rows (columns) of the matrix are linearly independent. This standard theorem from linear algebra provides a method for the generation of invertible matrices. If the rows (columns) of the matrix are chosen from a linearly independent set of vectors, invertibility necessarily follows. A method to accomplish this is easily realizable (in base two) once it is observed that the columns of the matrix A = encode base 2: $\{1, 2, \ldots, 2^{m-1}\}$ contain all possible coefficient combinations for distinct linear combinations of a set of n vectors, where $1 \leq n \leq m$. The columns of this matrix also form the set of all possible m-bit binary numbers (excluding zero). The properties of the matrix A allow it to be used both as a multiplier to form linear combinations and as a set (with the columns as its elements) from which the rows of the matrix being generated are chosen.

To illustrate the properties of matrix A, a three-by-three invertible matrix (F) will be generated. For m = 3,

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Two columns of A are then arbitrarily chosen to be the first two rows of F, with the remaining row being filled in by the zero vector. Choosing the first two columns of A yields

$$F = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

To complete F, it is necessary to compute all linear

[1] *M. G. Karpovsky, Finite Orthogonal Series in the Design of Digital Devices, Halstead Press, New York (1976), 100.*

* *Unless otherwise stated, all arithmetic operations in this paper are performed modulo two.*

combinations of the first two rows of F. This is achieved by forming the matrix product of A^T and F; i.e.,

$$A^T \otimes F = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} = G. \qquad (2)$$

The resultant matrix (G) has three distinct rows (excluding (0,0,0)), which are (0,1,0), (0,0,1), and (0,1,1). These three vectors are all the possible linear combinations of the rows of F. Deletion of these vectors from the columns of A produces a reduced matrix,

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

from whose columns the last row of F may be chosen. Choosing column four yields

$$F = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

which has a nonzero determinant, and thus an inverse matrix. Generalization of this method to higher-order matrices follows readily. A flowchart representing this method for a matrix of size m by m ($m \geq 3$) appears in figure 1 (p 7).

2. GENERATION OF S

Generation of an invertible matrix (S) which satisfies equation (1) is now a fairly straightforward matter. Since the product of S and $\vec{T}$ is the vector (0,..., 0, 1), the vector product of any of the first (m − 1) rows of S with $\vec{T}$ must equal zero, and the product of the last row of S with $\vec{T}$ must equal one. These requirements allow two submatrices of the matrix A = encode base 2: $\{1, 2, \ldots, 2^{m-1}\}$ to be formed, one from which the first (m − 1) rows of S are chosen, and another from which the last row of S is chosen. Set representations of these two matrices are

$$C = \{\text{columns of A} \mid a_i \odot \vec{T} = 1\} \text{ and}$$

$$D = \{\text{columns of A} \mid a_i \odot \vec{T} = 0\}$$

where $\odot$ represents the dot product modulo two, and $1 \leq i \leq (2^{m-1})$

2.1 *Method I*

A method of generating S, which is similar to the procedure outlined for the generation of an invertible matrix, appears in figure 2 (p 8). One need only choose (m − 1) linearly independent columns of D for the first (m − 1) rows of S, then choose a column of C, which is independent of these columns of D, for the last row of S.

Direct coding into APL of the algorithm presented in figure 2 is possible for $m \leq 13$. However, for $m > 13$, a 478096 byte workspace becomes filled, and the central processing unit (CPU) time necessary becomes excessive. The storage space necessary will be reduced if the way the data are represented in the computer is considered, and the CPU time will be reduced if the production of the linear combination matrix G (eq 2) is examined.

Operating modulo two allows all variables and operations performed to be converted from integer to logical. Addition and multiplication modulo two are converted to "not equals" and "and," respectively. This change from integer to logical representation produces a reduction in storage space. For example, on the IBM 370/168 (in VSAPL), 648 bytes are required to store a matrix with 155 integer elements, while storage of the same matrix with logical elements requires only 48 bytes. Thus, to decrease the storage space necessary, all operations and data will be logical.

The actual CPU time necessary to compute S can be greatly reduced by considering the production of the linear combination matrix (eq 2). This reduction is accomplished by decreasing the size of A^T and S. Considering the example where a

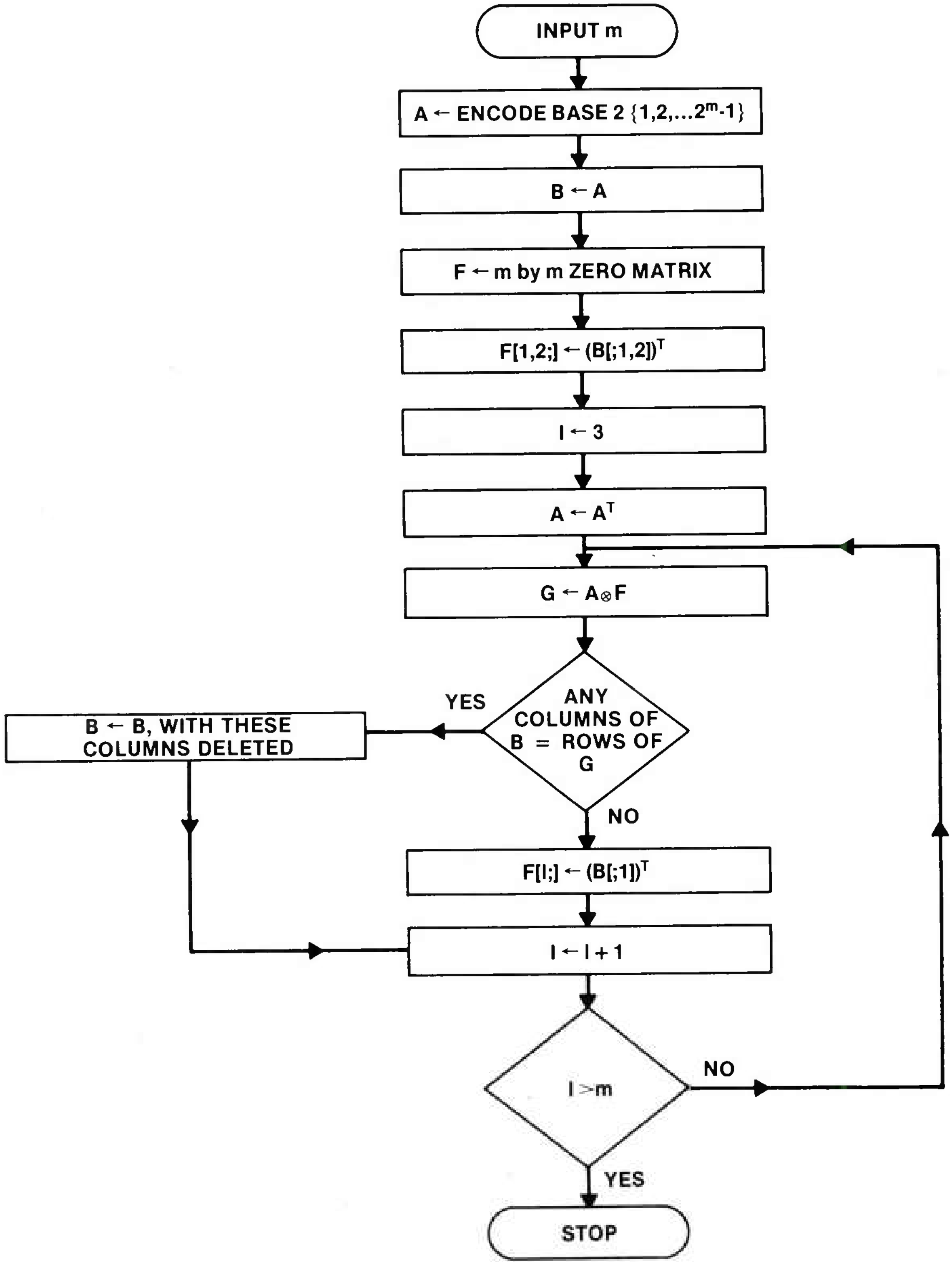

Figure 1. Generation of an Invertible Matrix.

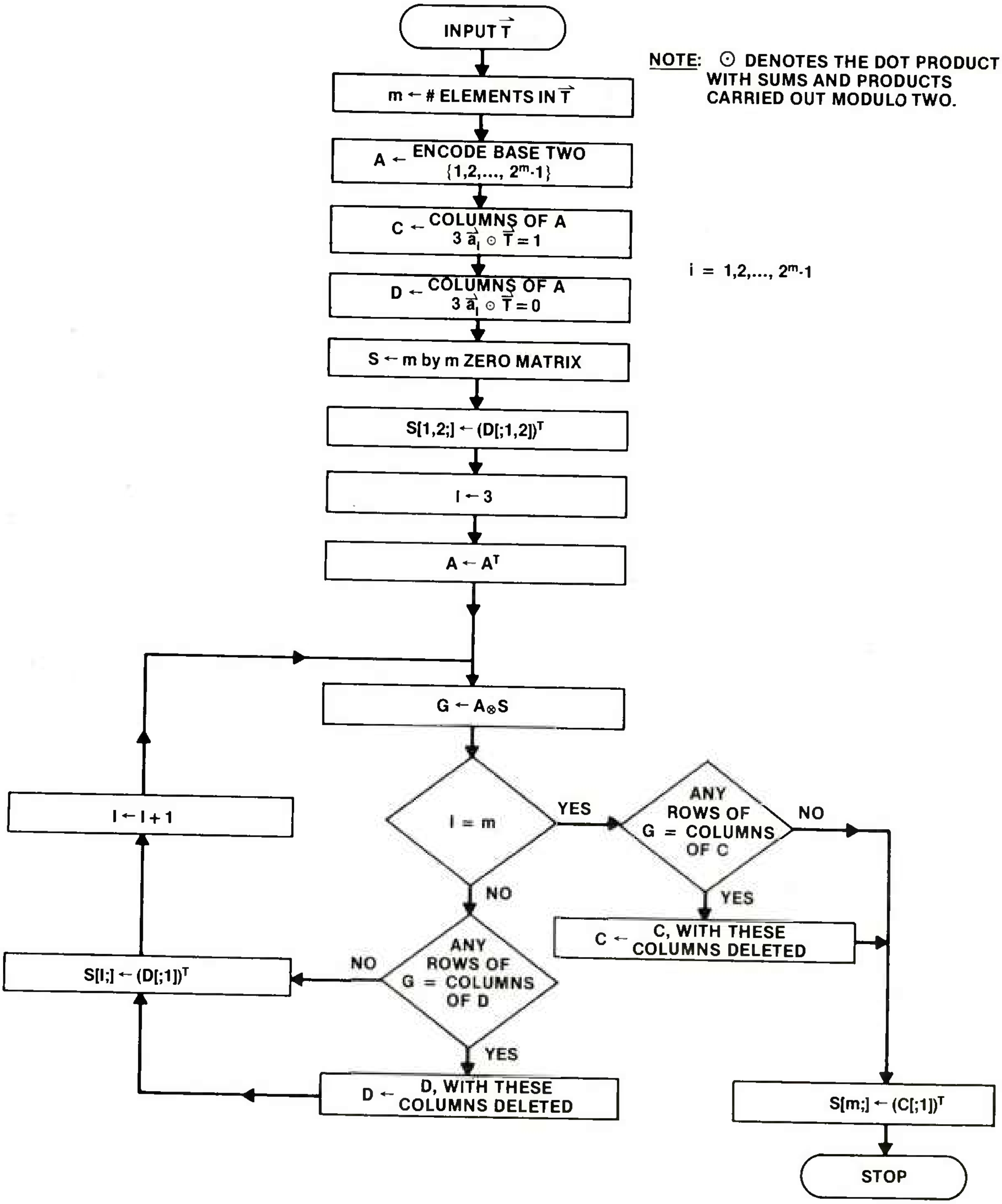

Figure 2. Method I: Generation of S.

3-by-3 matrix was generated, it is observed that the third column of A^T and the third row of F (eq 2) are not needed to compute G. Deletion of the third column of A^T gives

$$A^T = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$$

This size of this matrix may be further reduced by deleting all odd rows (which, except for row 1, are duplicates of the preceding even row); i.e.,

$$A^T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

The product of this matrix with the first two rows of F is

$$A^T \otimes F[1, 2] = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} = G' \qquad (4)$$

Comparing G' with the matrix G of equation 2 shows that the same basic result has been achieved, but with a considerable decrease in the number of operations performed.

Generation of an m-by-m matrix (S), requires the matrix A^T to have shape $(2^m - 1)$ by m. Thus, A has $(m2^m - m)$ elements. Removal of the last column of A^T leaves a matrix with rows which are not needed when computing the linear combination matrix G. Further removal of all odd rows of A^T (the rows which are not needed) yields a reduced matrix A'^T with shape $(2^{m-1} - 1)$ by $(m - 1)$, which contains $(m2^{m-1} + 1 - 2^{m-1})$ elements. This matrix A'^T has $[(2^{m-1}(m + 1)) - 1]$ elements less than the original A^T, and is the largest matrix necessary for the computation of G. Further reductions of A'^T, which depend upon the index of the row (of S) being filled, are also possible. A procedure for reducing A'^T when filling the Ith row of S follows.

(1) Delete all rows of A'^T except those with indices contained in (2^{m-I}) x $(1, \ldots, 2^{I-1} - 1)$, where x represents ordinary multiplication.

(2) Delete all but the first $(I - 1)$ columns from A'^T. Forming the product A'^T and the first $(I - 1)$ rows of S will now produce the desired matrix G'. Considering the case $m = 4$, the initial reduction of A^T produces

$$A'^T = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Letting $I = 3$, all rows of A'^T, except those with indices 2, 4, and 6, and all columns except the first $(I - 1) = 2$, are deleted, leaving

$$A'^T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

The product of A'^T and the first two rows of S may now be formed to produce the reduced G. Thus, for $I = 3$, the original matrix A^T has been reduced from a 15 by 4 to a 3-by-2 matrix, which decreases the total number of elements from 60 to 6.

When implemented in VSAPL, the algorithm in figure 2, with the inclusion of the optimization procedures just discussed, can handle up to 16 I/O bits.

2.2 *Method II*

A second method for generating S, which uses very little CPU time and storage space, is presented in this section. This method is programmable in VSAPL and can handle more than 1000 I/O bits.

Examination of the set D', which has the columns of D plus the zero vector as its elements, yields the observation that D′ forms a group with respect to $\oplus$. Thus, D′ is closed under $\oplus$, and the elements of C′, which are the columns of C, are independent of the elements of D′. From this property, it is observed that the last row of S may be filled by any vector whose product with T is 1.

The first $(m-1)$ rows of S are filled by generating $(m-1)$ m-bit vectors, which occupy different subspaces of m-dimensional space, and whose product with $\vec{T}$ is equal to zero. Since each of these vectors exists in a different subspace than each of the other $(m-2)$ vectors, they form a linearly independent set. A classic example of this is the "natural basis;" i.e., for $m = 3$, the natural basis is $\vec{E}_1$, $\vec{E}_2$, $\vec{E}_3$, or (1, 0, 0) (0, 1, 0), (0, 0, 1).

The first n rows of S are filled by vectors from the natural basis which correspond to the positions of the zeros in $\vec{T}$, where $\vec{T}$ contains n zeros and p ones with $(n + p) = m$. This leaves $(m - n - 1) = (p - 1)$ rows to fill (excluding the mth row). The vectors which are produced to fill these remaining rows contain two ones and $(m - 2)$ zeros. The first "one" always occurs in the bit occupied by the first one appearing in $\vec{T}$. The position of the second one ranges over all the bits occupied by the remaining $(p - 1)$ ones in $\vec{T}$. Since there are p ones and the position of the second one assumes $(p - 1)$ different positions, $(p - 1)$ vectors are generated. Thus, $(p - 1) + n = (m - 1)$ rows of S are filled. A suitable vector for the mth row of S is one which has a one in the bit occupied by the first one in $\vec{T}$, and zeros everywhere else. For example with $T = (1, 0, 1, 1, 0)$ the first two rows of S are

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(or $\vec{E}_2$ and $\vec{E}_5$); the third and fourth rows of S are

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \quad ;$$

and the fifth row of S is (1, 0, 0, 0, 0). Thus,

$$S = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{, and the}$$

product of S and $\vec{T}$ equals (0, 0, 0, 0, 1). A flowchart of this method appears in figure 3(p 11), and a program that implements it in VSAPL appears in appendix A (p 13).

3. SUMMARY

Two procedures have been developed for the generation of an invertible modulo two matrix which satisfies equation 1. Method I is based on an algorithm developed for the production of invertible modulo two matrices, while Method II produces the required matrix by a direct examination of the input vector $\vec{T}$. Both methods have been implemented in VSAPL, and the program for Method II, which can generate much larger matrices than the program for Method I, appears in appendix A. This program will generate matrices of size 1000 by 1000 (or with 100,000 elements) in a 478096 byte workspace.

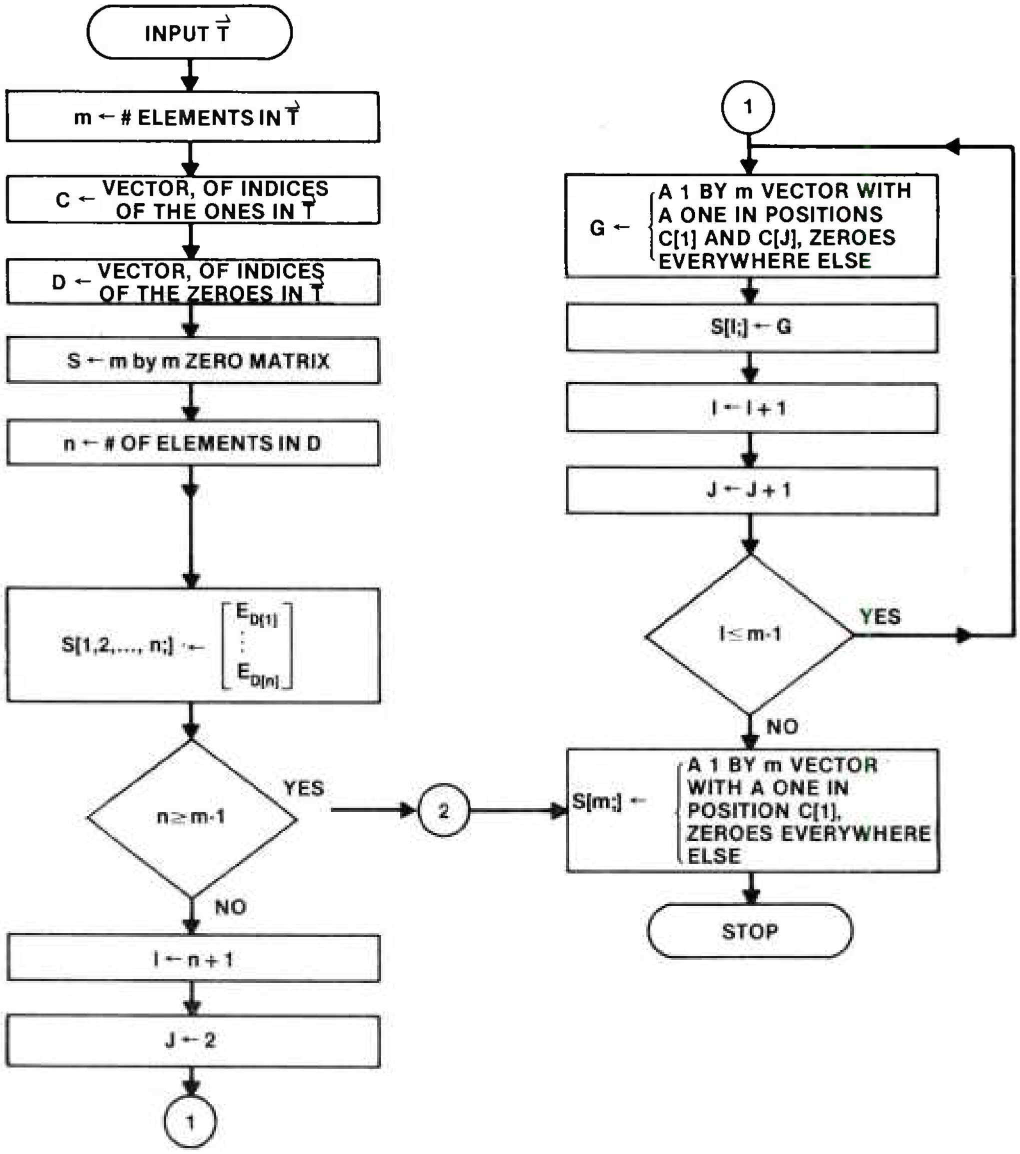

Figure 3. Method II: Generation of S.

APPENDIX A. — Program to generate Matrix S

A program that generates the matrix S (eq 1, body of report) is presented in figure A-1. The program is written in VSAPL, and was run on an IBM 370/168 computer. With a workspace of 478096 bytes, this program will produce matrices which are larger than 1000 by 1000. The program itself is a direct coding into VSAPL of the algorithm presented in figure 3 (body of report). The input to this program is the vector $\overline{T}$ (as a row vector) and the output is the matrix S.

```
∇SIGMA [⎕] ∇
      ∇ S←SIGMA T;A;C;D;E;G;I;J;M;N
[1]   ⍝ THIS FUNCTION GENERATES AN INVERTIBLE MATRIX WHICH
[2]   ⍝ SATISFIES KARPOVSKY'S MATRIX EQUATION.
[3]    M←⍴T
[4]    A←⍳M
[5]    C←T/A
[6]    D←(~T)/A
[7]    N←⍴D
[8]    S←(M,M)⍴0
[9]   ⍝ FILLING THE FIRST N ROWS OF S WITH VECTORS FROM
[10]  ⍝ THE NATURAL BASIS .
[11]   E←D∘.=A
[12]   S[⍳N;]←E
[13]   →(N≥M-1)/LAST
[14]   I←N+1
[15]   J←2
[16]  ⍝ ENTERING A LOOP WHICH FILLS ROWS N+1 TO M-1.
[17]  HIT:G←∨/[1](C[1,J]∘.=A)
[18]   S[I;]←G
[19]   I←I+1
[20]   J←J+1
[21]   →(I≤M-1)/HIT
[22]  ⍝ FINISH LOOP .
[23]  ⍝
[24]  ⍝ FILLING LAST ROW OF S.
[25]  LAST:S[M;]←(C[1]=A)
      ∇
```

Figure A-1. Program that generates matrix S (eq 1, body of report).

DISTRIBUTION

ADMINISTRATOR
DEFENSE DOCUMENTATION CENTER
ATTN DDC-TCA (12 COPIES)
CAMERON STATION, BUILDING 5
ALEXANDRIA, VA 22314

COMMANDER
US ARMY RSCH & STD GP (EUR)
ATTN LTC JAMES M. KENNEDY, JR.
CHIEF, PHYSICS & MATH BRANCH
FPO NEW YORK 09510

COMMANDER
US ARMY MATERIEL DEVELOPMENT & READINESS COMMAND
ATTN DRXAM-TL, HQ TECH LIBRARY
5001 EISENHOWER AVENUE
ALEXANDRIA, VA 22333

COMMANDER
US ARMY ARMAMENT MATERIEL READINESS COMMAND
ATTN DRSAR-LEP-L, TECHNICAL LIBRARY
ROCK ISLAND, IL 61299

COMMANDER
US ARMY MISSILE & MUNITIONS CENTER & SCHOOL
ATTN ATSK-CTD-F
REDSTONE ARSENAL, AL 35809

DIRECTOR
US ARMY MATERIEL SYSTEMS ANALYSIS ACTIVITY
ATTN DRXSY-MP
ABERDEEN PROVING GROUND, MD 21005

DIRECTOR
US ARMY BALLISTIC RESEARCH LABORATORY
ATTN DRDAR-TSB-S (STINFO)
ABERDEEN PROVING GROUND, MD 21005

TELEDYNE BROWN ENGINEERING
CUMMINGS RESEARCH PARK
ATTN DR. MELVIN L. PRICE, MS-44
HUNTSVILLE, AL 35807

COMMANDING OFFICER
NAVAL TRAINING EQUIPMENT CENTER
ATTN TECHNICAL LIBRARY
ORLANDO, FL 32813

US ARMY ELECTRONICS TECHNOLOGY & DEVICES LABORATORY
ATTN DELET-DD
FORT MONMOUTH, NJ 07703

AMES LABORATORY (ERDA)
IOWA STATE UNIVERSITY
ATTN MATHEMATICAL & COMPUTER SCIENCES ACTIVITY
AMES, IA 50011

DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS
ATTN LIBRARY
WASHINGTON, DC 20234

DIRECTOR
DEFENSE COMMUNICATIONS AGENCY
ATTN TECH LIBRARY
WASHINGTON, DC 20305

DIRECTOR
DEFENSE COMMUNICATIONS ENGINEERING CENTER
ATTN TECHNICAL LIBRARY
1860 WIEHLE AVE
RESTON, VA 22090

COMMANDER-IN-CHIEF
CONTINENTAL AIR DEFENSE COMMAND, JCS
ATTN TECHNICAL LIB
ENT AFB
COLORADO SPRINGS, CO 80912

COMMANDER-IN-CHIEF
US EUROPEAN COMMAND, JCS
ATTN TECHNICAL LIBRARY
APO NEW YORK, NY 09128

DIRECTOR
DEFENSE NUCLEAR AGENCY
ATTN TITL, LIBRARY
WASHINGTON, DC 20305

COMMANDER
US ARMY AVIATION RESEARCH & DEVELOPMENT COMMAND
ATTN DRDAV-N, ADVANCED SYS TECHNOLOGY INTEGRATION
P.O. BOX 209
ST LOUIS, MO 63166

COMMANDER
US ARMY COMPUTER SYSTEMS COMMAND
ATTN TECH LIB
FORT BELVOIR, VA 22060

COMMANDER
US ARMY MISSILE RES & DEV COMMAND
ATTN DRDMI-T, SCI & ENGR ADVISOR
ATTN DRDMI-TB, RESTONE SCIENTIFIC INFO CENTER
ATTN DRDMI-C, ADVANCED SYS CONCEPTS OFC
REDSTONE ARSENAL, 35809

DISTRIBUTION (Cont'd)

COMMANDER
NAVAL OCEAN SYSTEMS CENTER
ATTN RESEARCH & TECH OFFICE
ATTN ENGINEERING & COMPUTER SCIENCE
SAN DIEGO, CA 92152

DIRECTOR
NAVAL RESEARCH LABORATORY
ATTN 5300, RADAR DIV
WASHINGTON, DC 20375

COMMANDER
NAVAL SEA SYSTEMS COMMAND HQ
DEPT OF THE NAVY
ATTN NSEA-09G32, TECH LIB
ATTN SEA-62R13
WASHINGTON, DC 20362

COMMANDER
NAVAL SHIP R&D CENTER
ATTN COMPUTATION MATHEMATICS DEPT
BETHESDA, MD 20034

COMMANDER
NAVAL SURFACE WEAPONS CENTER
ATTN DF-36, RADAR SYS ENGR BR
ATTN DX-21, LIBRARY DIV
DAHLGREN, VA 22448

COMMANDER
NAVAL SURFACE WEAPONS CENTER
ATTN WX-40, TACHNICAL LIB
WHITE OAK, MD 20910

SUPERINTENDENT
HQ US AIR FORCE ACADEMY
ATTN TECH LIB
USAF ACADEMY, CO 80840

DIRECTOR
AF AVIONICS LABORATORY
ATTN LDF (TSR), STINFO BR
WRIGHT-PATTERSON AFB, OH 45433

COMMANDER
AF WEAPONS LAB, AFSC
ATTN AD, COMPUTATIONAL DIV
KIRTLAND AFB, NM 87117

AMES RESEARCH CENTER
NASA
ATTN TECHNICAL INFO DIV
MOFFETT FIELD, CA 94035

DIRECTOR
NASA
GODDARD SPACE FLIGHT CENTER
ATTN 250, TECH INFO DIV
GREENBELT, MD 20771

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
ATTN TECHNICAL LIBRARY
4800 OAK GROVE DRIVE
PASADENA, CA 91103

DIRECTOR
NASA
LANGLEY RESEARCH CENTER
ATTN TECHNICAL LIBRARY
HAMPTON, VA 23665

US ARMY ELECTRONICS RESEARCH
& DEVELOPMENT COMMAND
ATTN TECHNICAL DIRECTOR, DRDEL-CT

HARRY DIAMOND LABORATORIES
ATTN 00100, COMMANDER/TECH DIR/TSO
ATTN CHIEF, DIV 10000
ATTN CHIEF, DIV 20000
ATTN CHIEF, DIV 30000
ATTN CHIEF, DIV 40000
ATTN RECORD COPY, 81200
ATTN HDL LIBRARY, 81100 (3 COPIES)
ATTN HDL LIBRARY, 81100 (WOODBRIDGE)
ATTN TECHNICAL REPORTS BRANCH, 81300
ATTN CHAIRMAN, EDITORIAL COMMITTEE
ATTN LAWSON, W., 34400
ATTN JOHNSON, R., 15200
ATTN JEDREY, T.C., 15200 (10 COPIES)